Commodore

The Transactor

Commodore Canada's
Tech/News Periodical

VOLUME 3
Issue #2

Bits & Pieces

Disabling The STOP Key

Here are a couple of ways to disable the 4.0 STOP key. If you don't need the internal clock:

```
Disable with    POKE 144, 88
Restore with    POKE 144, 85
```

Notice that to disable you simply add three to low order address of the IRQ vector, as with all other versions of Commodore Basic.

If you do need the clock, Jim Butterfield has a tidy little Basic routine that does it:

```
100 D$ = "20>:??:9??8=9;004<58>4"
110 FOR J = 1 TO LEN (D$)/2
120 POKE J+852, ASC(MID$(D$,J*2-1))*16
    + ASC(MID$(D$,J*2))-816
130 NEXT
```

WordPro and The 40-8032

WordPro 3 Plus will work on the new 12" screen 4032 given these POKE conversions:

```
1. LOAD "WORDPRO 3+
2. POKE 6084, 36
3. POKE 6088, 24
4. POKE 6090, 147
5. POKE 2438, 108
6. POKE 2439, 252
7. POKE 2440, 255
```

While on the subject of WordPro, several have asked "Can you get a SEQuential file into WordPro?". The answer is yes, but if the method seems a little abnormal, that's because it is!

First type about a dozen variable blocks (Control P). For now we'll put one on each line. Now using the "Line Duplicate" function, fill the whole worlspace with variable blocks and Memorize this for future use.

Index Transactor #2

Bits & Pieces 1
  Disabling STOP 1
  WordPro & The 40-8032 1
  CBM ROMs 3
CBM 8010 Notes 4
PET Show Report 6
40/80 Video Interface 11
Dial-A-ROMs 29
Joystick/Keyboard Routine 30
What Hardware Do I Have? 34
The Pat 40 37
A First Look At The SUPERPET 40

Pick some SEQ file that you would like to edit using WordPro. The idea here is to make WordPro "think" that you're going to use the SEQ file data to insert into a form letter which, in this case, is just blank. The printer must be connected. Type:

Control 'O'utput + 'S' <return>

WordPro will prompt you for the name of the SEQ file. Type it in and hit return. The variable blocks will begin to fill up with the data from the SEQ file. However, if the last variable block is filled with data, WordPro will procede to Output to the printer. Hitting STOP halts output but then WordPro goes and erases the contents of you're variable blocks... ARGHH! Solution?.. Always have at least one more variable block in WordPro text than you have carriage returns in you're SEQ file. This way WordPro won't be able to find enough data to fill all the blocks and reports an error, at which point you're data is left in tact.

You may have to keep adding variable blocks to the point where the SEQ file is exhausted before the last block. This may require more than one block on a line. However, a sequential output to disk will concatenate blocks on the same line which may not be too desirable. To delimit blocks with carriage returns, you must either keep each item on its own line or use the programmable character facility in WordPro. Define some character, say 1, equal to ASCII 13 ('tick'1=13) and insert Control:1s between the blocks. See WordPro manual for more on defining special characters.

Commodore ROMs

Retrofit ROM sets are available from your dealer for upgrade to BASIC 4.0 and/or DOS 2.0. Cost for each set is (suggested retail) $100.00.

The manual that is supplied with the 8010 modem is fairly good, but I found I still had to read between the lines to really get it right.

My objective was to be able to have two PETs connected by modem, that could freely talk with each other. I intended to implement this by having a program in each PET which sends characters typed in to the modem, and echoes them to the screen, while displaying any received characters in reverse to distinguish them, from the operators text.

The first program that I wrote was based on what I had read in the manual, that if ST (the status variable) is zero, a character is pending and so I wrote a program like this:-

```
100 OPEN 5,5
110 GET#5,A$:IF ST<>0 THEN PRINT"[RVS]"A$"[OFFRVS]";
120 GETA$:IF A$<>"" THEN PRINT#5,A$;:PRINT A$;
130 GOTO 110
```

This program was totally unacceptable, because it dropped 50% of all transmitted characters. However I had read about the SRQ method of using the modem. SRQ is the main reason why the PET IEEE is not a real IEEE-488 bus. It is a real line on the bus in the PET, but is not used. SRQ stands for service request, and all it does is to allow a peripheral to tell a controller that it requires servicing. So the second program:-

```
100 OPEN 5,5
110 IF PEEK(59427)AND128 THEN GET#5,A$
    :POKE 59426,0:PRINT"[RVS]"A$"[OFFRVS]";
120 GETA$:IF A$<>"" THEN PRINT#5,A$;:PRINT A$;
130 GOTO 110
```

This worked a treat. Line 110 (which is the important one), says "IF <the modem needs servicing> then <service it>:<tell it, it has been serviced>:<display character from modem>".

Once I had discovered how easy it was to communicate via the 8010, I decided to add a little style. And so the third program evolved from my fingertips 20 minutes later. This program uses the window facility of the 8032, and so it will not work on 40 column machines I'm afraid. The top window displays characters typed locally and the bottom window shows remote activity. I plan to add peripheral transmit and receive file capability, whereby any device can be toggled on or off. This will be published in a future Transactor.

8032 Dual Window Modem Communicator

```
1000 OPEN 5,5:PRINT"[HM HM CLR]"
1010 PRINT"[CLR]"
1020 FOR I=0 TO 79:POKE 33728+I,64:NEXT
1030 CD$="[DN DN DN DN DN DN DN DN DN DN DN DN]"
1040 X1=0:Y1=0:X2=0:Y2=0
1050 SI=59427:MS=128:IO=59426:ZE=0
1060 TL=224:ML=13:CP=198:CL=216:BL=225:HF=232
1070 LL=80:S1=32768:S2=33808:US=127:BP=11
1080 POKE S1,160:POKE S2,160:ES$=CHR$(27)
2000 IF NOT PEEK(SI) AND MS THEN 3000
2010 GET#5,A$:POKE IO,ZE:PRINT"[HM HM]";:POKE HF,ZE:POKE TL,ML
2020 P=S2+X2+Y2*LL:POKE P,PEEK(P)ANDUS
2030 PRINT"[HM]"LEFT$(CD$,Y2)SPC(X2):POKE HF,ZE:PRINTA$ES$;
2040 X2=PEEK(CP):Y2=PEEK(CL)-ML
2050 P=S2+X2+Y2*LL:POKE P,PEEK(P)ORMS
3000 GETA$:IF A$="" THEN 2000
3010 PRINT"[HM HM]";:POKE BL,BP:POKE HF,ZE
3020 P=S1+X1+Y1*LL:POKE P,PEEK(P)ANDUS
3030 PRINT"[HM]"LEFT$(CD$,Y1)SPC(X1):POKE HF,ZE:PRINTA$ES$;
3040 X1=PEEK(CP):Y1=PEEK(CL)
3050 P=S1+X1+Y1*LL:POKE P,PEEK(P)ORMS
3060 PRINT#5,A$;:GOTO 2000
```

The Second
International Commodore PET Show

Paul Higginbottom
CBM Software Dept.

This was the second PET show, and was much larger then the first. There were ninety three stands comprising of dealers, software houses, electronics specialists, and systems houses from all over the world. The products on show fall into two catagories, namely software, and hardware packages. Each will be discussed below.

Software Packages

The software packages being shown were mostly accounting packages. Apart from accounting, there were a couple of payroll packages, a few data bases, quite a few communications packages, and other more esoteric areas of software. And of course, the wordprocessing packages WORDPRO, and WORDCRAFT.

Accounting software

The accounting packages were all good, giving various combinations of the following features:-

-Sales ledger
-Purchase ledger
-Nominal ledger
-Integrated ledgers
-Open item ledgers
-Invoicing
-Integration with stock control

I.S.A Computer Sevices in Bradford, Yorkshire who already have their package 'THE ACCOUNTANT', sold by Commodore in the U.K, showed me their complete range (contact name: Ken Marchant). This was very impressive, and since I went up to Bradford for a week while I was working with Commodore U.K, I am aware of their setup, which is also very impressive. They are a data preparation agency as well as a software house with about sixty staff. They have about ten programmers on staff, and are well organized because of their systems experience.

Bartholemews Business Systems (contact name: Bob Hawthorn) who are a Commodore dealer in the U.K were showing a new ledger package that had been taken off a PDP-11 system. This looked good, but I felt that because this show was its launch, it would need field testing before it could be considered.

As I have said, there were many accounting packages at the show, and I haven't got time to cover them all here. I.S.A looked the best because of it comprehensive coverage of different options, and the functioning of the package. It also says something about I.S.A when they use their package for their own accounting!

Payrolls

As for payrolls, well they cannot be reviewed here due to major differences in tax structure.

Databases

Bristol Software Factory of 'OZZ' fame, who are a leading software house in the PET world (contact name: John Kyle-Price), were exhibiting a new package called 'SILICON OFFICE'. They said "No other package on any computer in the world can offer you as many facilities as Silicon Office" which I felt was optimistic. Anyway, it is a package that has been designed to incorporate data processing, word processing, and communications software into one package. Firstly, it should be noted that this software will only run on the CBM 8096 (a future product from Commodore which is an CBM 8032 with an extra 64K bank switchable RAM) and thus the package will not be available until at least September. It is based around the OZZ package, but does not require the user already own OZZ since it IS OZZ with a whole array of new features. It is now a fully programmable database, allowing 'OFFICE' programs to be written by the user, or probably more likely, set up by a dealer for the user. It has a built in wordprocessor which can cope with multiple columns of text, expressions to be evaluated out of a data file such as TOTAL VALUE OF STOCK or DEALER PRICE=RETAIL-RETAIL*.3, right justification, automatic centering of text, virtual memory processing for page overflow, stored paragraph or page recall and insertion, horizontal scrolling of text for pages wider than eighty columns, and many other features. However, it seemed to lack the completeness of functions that other dedicated wordprocessing packages have. I thing I really liked was the fact that their were no spurious characters embedded in the text like, left arrows, reverse field text, graphics symbols etc. as with WORDPRO, and WORDCRAFT to a lesser extent. The operation of it worked more on the basis of the format of the text, rather than where certain symbols were. For example, an empty line of text, or an indented word at the start of a line, signified that the previous line was the last line of a paragraph. The communications software contained intelligent terminal modes driving a modem, with 'Silicon Office' protocols which means that it is probably not very compatible with much else, except another Silicon Office terminal.

They claim that this package will cope with just about any application. My own opinion is that what they have tried to do is an immense task and although it is desirable to try to cover many applications with just one package, what usually happens (and has happened here I think) is that although it may cover a wide range of applications, it does each one in a cumbersome fashion. If I bought 'Silicon Office', it would be for simple record storage, with selective mailshot capabilities on the wordprocessor, and

because I can have any number of remote work stations that can communicate with each other to produce global reports of (for example) a total inventory report of all work stations, or a current back order situation of a network of 'Silicon Office' stations.

On a final note about this package, I feel that this is a real breakthrough in software for Commodore computers, because of the amount of power contained in one package. This could be make or break time for our applications horizon; whether our computers should stick to smaller, more humble tasks, leaving larger applications to larger systems, or venture onward beyond the capabilities of 'SILICON OFFICE'.

Another database product on show was 'DMS - Data Management System' from Compsoft (contact name: Nick Horgan). This package has been around for a long time, and was one of the first of the programmable data base products. It is a comprehensive database with calculation facilities, selective reports and utilities to produce files of insertions into text, for both 'WORDPRO' and 'WORDCRAFT'. The length of time this product has been on the market and the popularity of it in the U.K must be give credit to this software house. My only reservations with 'DMS' are speed, however record access is quick (two seconds I am told) and speed is not necessary in practice; only in a demonstration!

Other database products on show were 'PETAID' (Stage One Software), 'FILEPROG' (Amplicon Microsystems), 'OZZ' (Bristol Software Factory), 'THE MANAGER' (BMB Compuscience).

Database software has come a long way since the dawn of the PET era, but they still seem to lack the elegance of something like 'VISICALC', and the wordprocessing packages such as 'WORDPRO' and 'WORDCRAFT'. 'The Manager' which is available from Commodore Canada, seems to be the most transparent database product when configured for an application. What I mean is that it seems less like a database, and more like an applications package, when configured, than the others. 'SILICON OFFICE' will be by far the most powerful database product available in the world, when it becomes available.

Communications Software

There were a good number of communications packages which proves the potential of this marketplace. Rod Welburn is Communications Manager of Commodore U.K and he feels that there is a huge potential in using the CBM 8032 as an intelligent terminal. This is true, primarily because of the price of the 8032 as compared to any terminal (dumb or smart). Provided the software is comprehensive and flexible enough to cope with the numerous protocols involved with 'talking' to different machines, Commodore could be onto a winner.

CORTEX (contact name: Howard Johnson) are a software house which deals only in communications software and a package of theirs called 'COMMUNICATOR' is a Commodore product in the U.K. The software they had on show was very good, especially due to the fact that their package allows a CBM to communicate with an IBM computer!

Kingston Computers is a Commodore dealer and deal primarily with network systems for CBM computers. Their product NETKIT allows any number of CBM's to talk to one another, and/or a mainframe.

Their were other packages at the show, but these need not be covered.

Other Software

Claremont Controls (contact name: Roy Stevenson) is a Commodore dealer who specialise in constructional software. They are the best in this field.

Final note on software

Of course, 'WORDPRO', and 'WORDCRAFT' were both on show. They were side by side, which was interesting in itself, and it seemed as though each stand was trying to print more 'demo' letters, than the other!

Hardware

Apart from numerous interfacing gadgets, D to A, and A to D converters and multi-port boards, the big steals of the show were the hard disk drive, and a CPM operating system.

MATOR HARD DISK - This is a 10, 22, or 35 megabyte hard disk. The DOS they are writing for it will be compatable with existing Commodore disk units. It was connected to the PET by an RS-232 interface at the show, but there will also be an IEEE version.

CPM system - For those not familiar with CPM, it is a small operating system less powerful than the CBM Kernal, but immensely popular on 808X, and Z80 machines. Because it has been around a very long time, there is a mountain of pre-packaged, good quality software for it. This implementation for the CBM is by means of a 'black box' that is attached to the memory expansion port, which contains a Z80 micropocessor.

HI-RES Board for the 8032 from Commodore U.K - An excellent product. It is a flexible graphics system with good supporting software to go with it. It allows pixel graphics on a matrix up to 512 X 512.

General

In general, I felt that the show had an enormous enthusiasm from both visitors, and exhibitors. Commodore's own stand was massive, and well presented. There was a VIC-20 ARCADE, which consisted of rows of VIC's lined up with joysticks, and televisions above, sealed in a presentation casing to prevent theft. These had games on them which were exact replicas of familiar arcade games such as 'PACMAN', 'INVADERS', 'GALAXIAN', and 'RALLY-X'. This was contantly occupied by thousands of children (big and small!). To the right of the arcade, along one wall, was all the packaged software products, and along the other wall was all the hardware products including the CBM cash register.

Jim Butterfield (I had to mention him somewhere), was mobbed wherever he went, and his seminars had five hundred people in a room big enough to hold fifty.

I felt that the level of expertise was high but still there are so few 'visicalc-wordpro-wordcraft' type of products, which are the kind that really do well. Accounting packages that are taken off larger machines is definitely a good approach to getting a system that incorporates the necessary options, and flexibility.

As the PET world gets ever bigger, our expectations of where it will all end, get blown apart each year.

Probably by next year, Commodore will use the Albert Hall for the show!

Video Interface
For 40/80 Column PET/CBMs

Ted Evers, Toronto
Metro Separate School Board

The demand for a larger display for demonstration purposes and classroom use, prompted a search for a video interface between the PET and an external monitor, of which we had a plentiful supply in our school system.

As there were several designs already available in many publications, we decided on a modified version, of a circuit, originally published by R. S. McLean, MECA, OISE.

The main design criteria of the system was, it had to:

1. be simple to build, install and wire to the mainboard
2. be easy to service
3. have no interference with other plug-in devices
4. be compatible with both the high-base and low-base type chassis
5. have standard output-connectors for both video and audio
6. accomodate our "DUMP" system at a minimum of extra cost, when so required.

To meet the set requirements, we decided on a printed circuit board design, measuring 2 1/2" x 3", this to be mounted by means of one #8 machine screw on the inside of the rear base apron, just above the user- and cassette-ports. A template was laid out for the hole locations to accommodate the BNC type video connector, the RCA type audio connector, the mounting hole for the P.C.B. and if so required, a DIN type connector for our "DUMP" system.

The circuit diagram is fairly straightforward. Four of the six inverter/drivers available in the 7406 I.C. are used in the video interface to combine the vertical and horizontal drive pulses with the video signal to obtain the required composite video.

The fifth inverter isolates the CB-2 line from the outside world, while the "DUMP" is activated by the last inverter. The input of this driver is connected to "E" (SAVE) of the cassette-port from the master-computer, while the output goes to "D" (LOAD) of up to 20 slave-computer units. Removing R8 and grounding "DUMP"-input will deactivate this feature.

By using an additional driving unit ("DUMP-MATE"), up to 60 slaves can be loaded at will.

A minor board modification and component change will make the design suitable for the CBM 8032 video interface. However, the audio will have to be connected directly to the CB-2 line and no dump is possible.

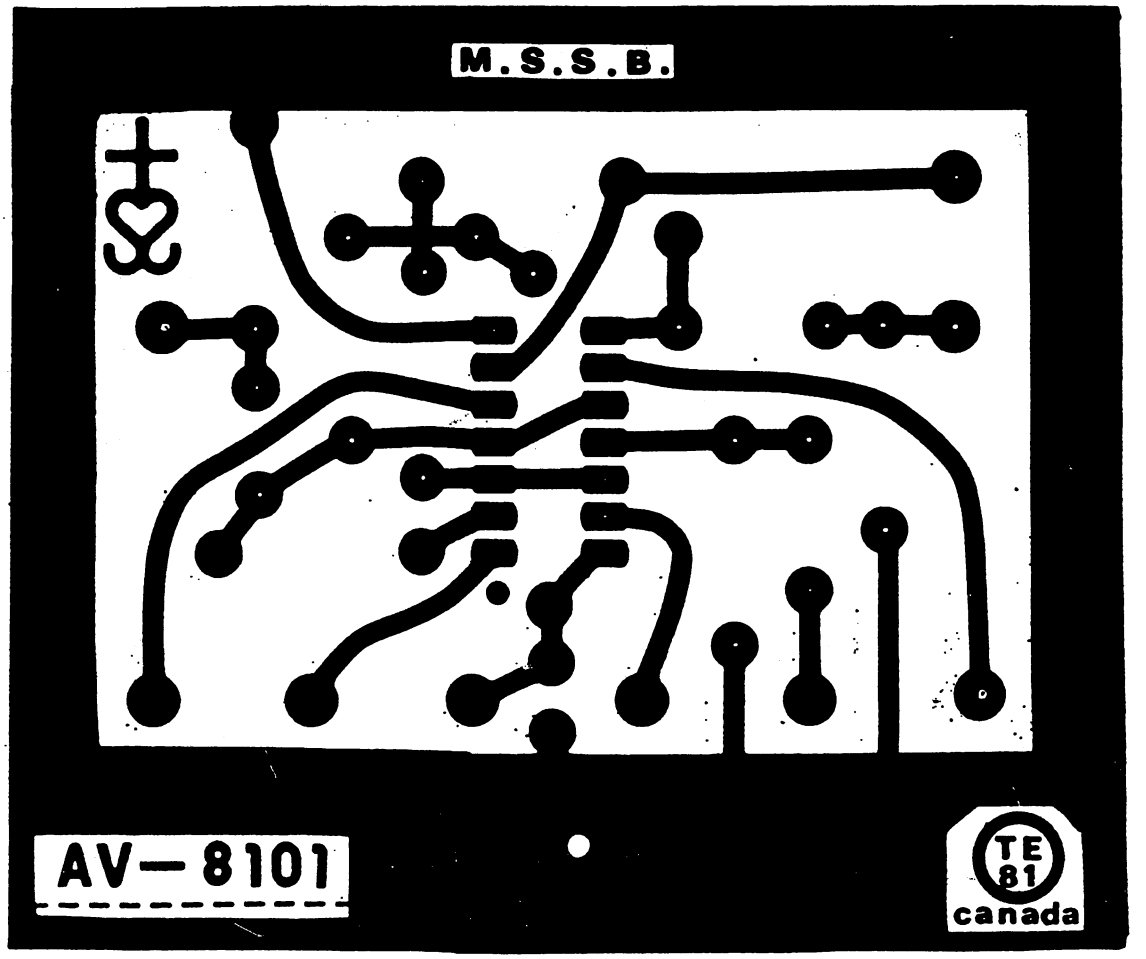
Editor's Note

The "DUMP" system mentioned is an interface that allows 20 slave PETs to download from a master PET using the 1st cassette port. "DUMP-MATE" boosts slave capacity to 60! Both circuits will appear in the next Transactor with modifications to make it compatible with the 8032 and the new "fat" 4032 which, by the way, also uses the 8032 video interface.

Included in the following diagrams is some circuitry for making Electrohome ETV-6 and ETV-7 video monitors switch-compatible for either 40 or 80 column interfacing. If operation with <u>both</u> 40 and 80 columns is'nt necessary, then simply adjust the proper potentiometers in the ETV-6/7 and change the screen voltage to the horizontal output tube to obtain the proper width. 80/fat40 column operationmay require this to bring down the vertical and horizontal dimensions, however with the 9" 40 you may not need it.

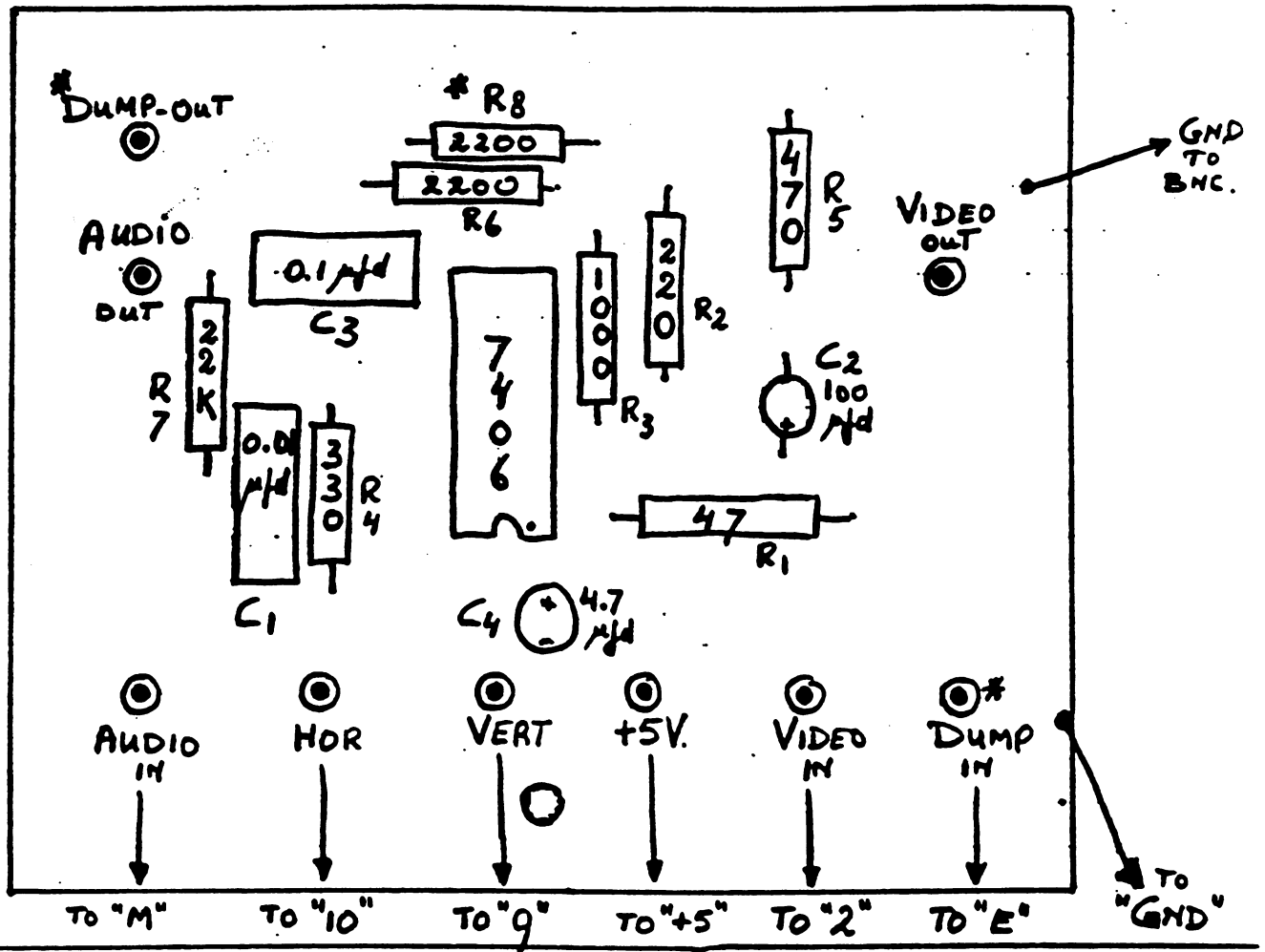
The 8032 interface board has a wire running along side the 7406. This connects pin 1 to pin 6 of the 7406. To get a "picture" of the final product, try super-imposing the component layout diagrams onto the <u>mirror image</u> of the artwork for the printed circuit board.

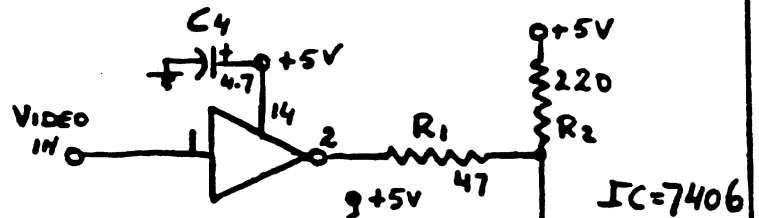
3.00 inches.

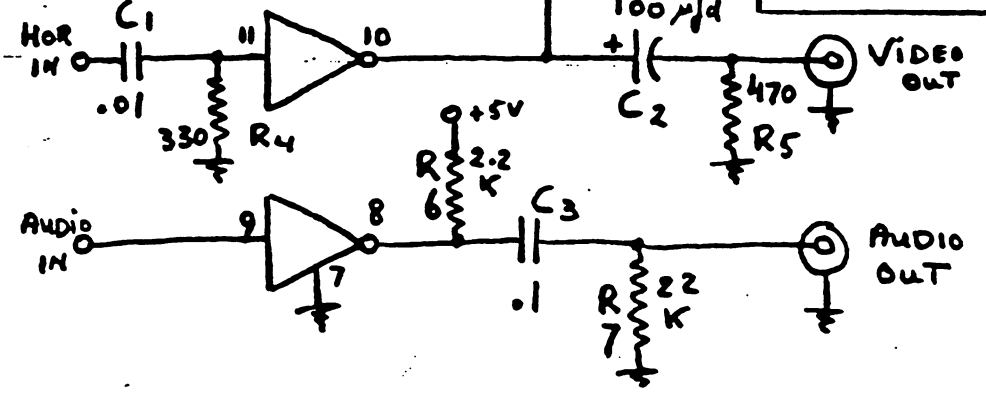


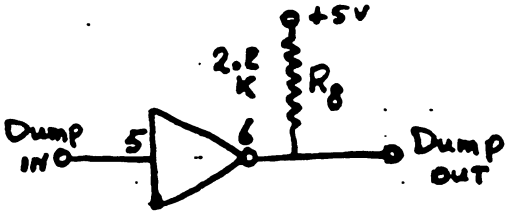
3.00 inches.

AV-8101 INTERFACE BOARD (Pet 2001). -2001-



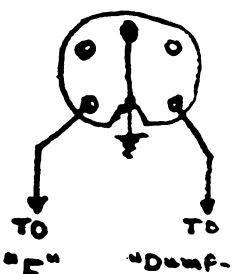






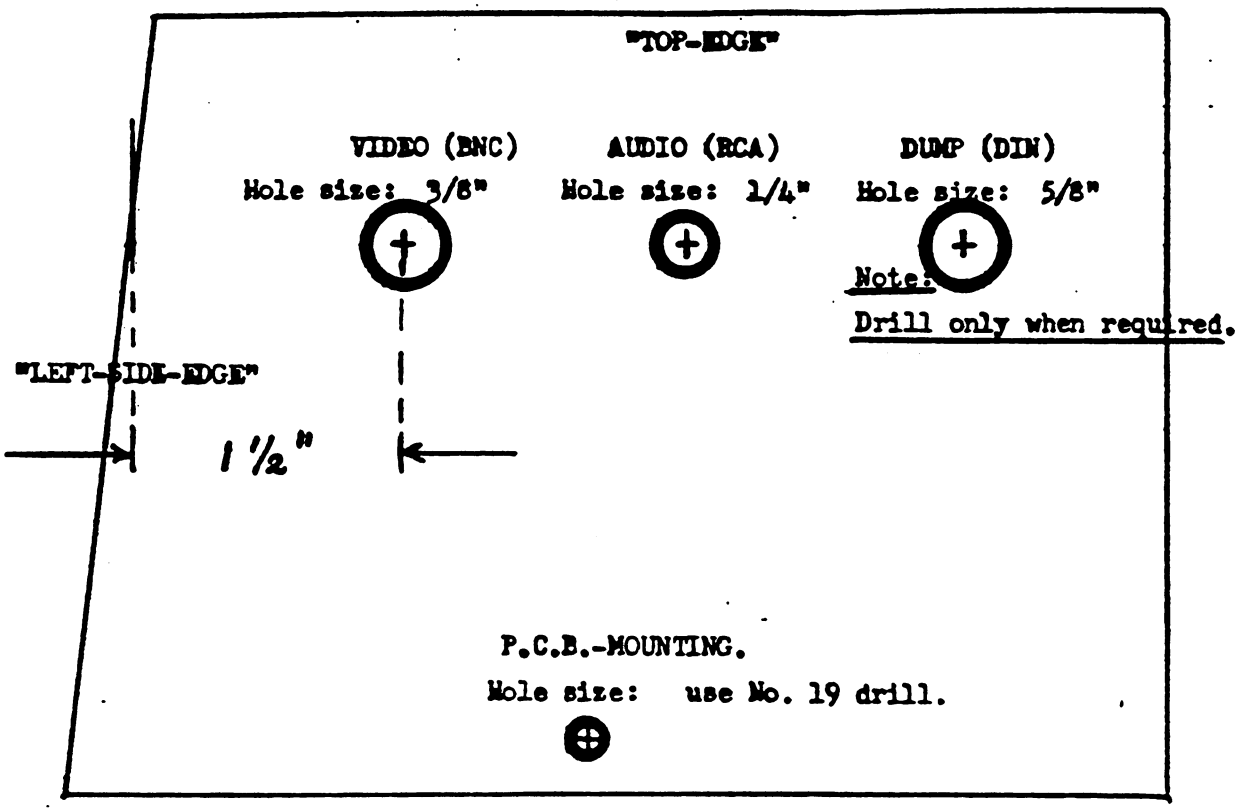
*note: When "Dump" not used, remove R8 and ground "Dump-in"

"Dump" DIN-Socket (Solder-view)



AV-8101
-2001-

PET 2001 REAR APRON DRILLING TEMPLATE

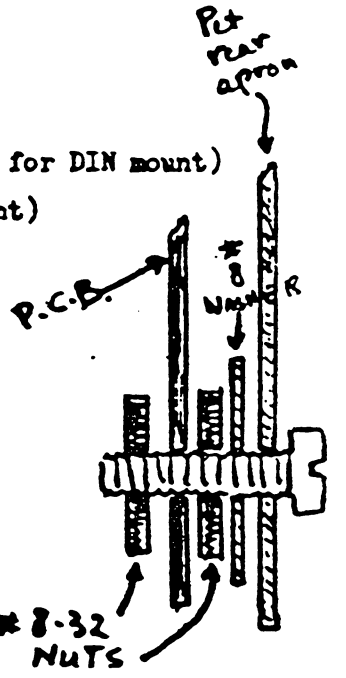


Cassette-port.

Parallel-user-port.

COMPONENT-REQUIREMENTS:

BNC-CONNECTOR	:	Amphenol 31-236 (UG/U-625B)
RCA-CONNECTOR	:	Switchcraft 3501-FP
GROUND-LUG	:	Amphenol 31-759
DIN-CONNECTOR	: *	Preh 71200-050
HARDWARE	: * 2	4-40 machine screws (1/2" for DIN mount)
	* 2	4-40 nuts (for DIN mount)
	1	8-32 X 1" machine screw.
	1	No. 8 washer.
	2	8-32 nuts.



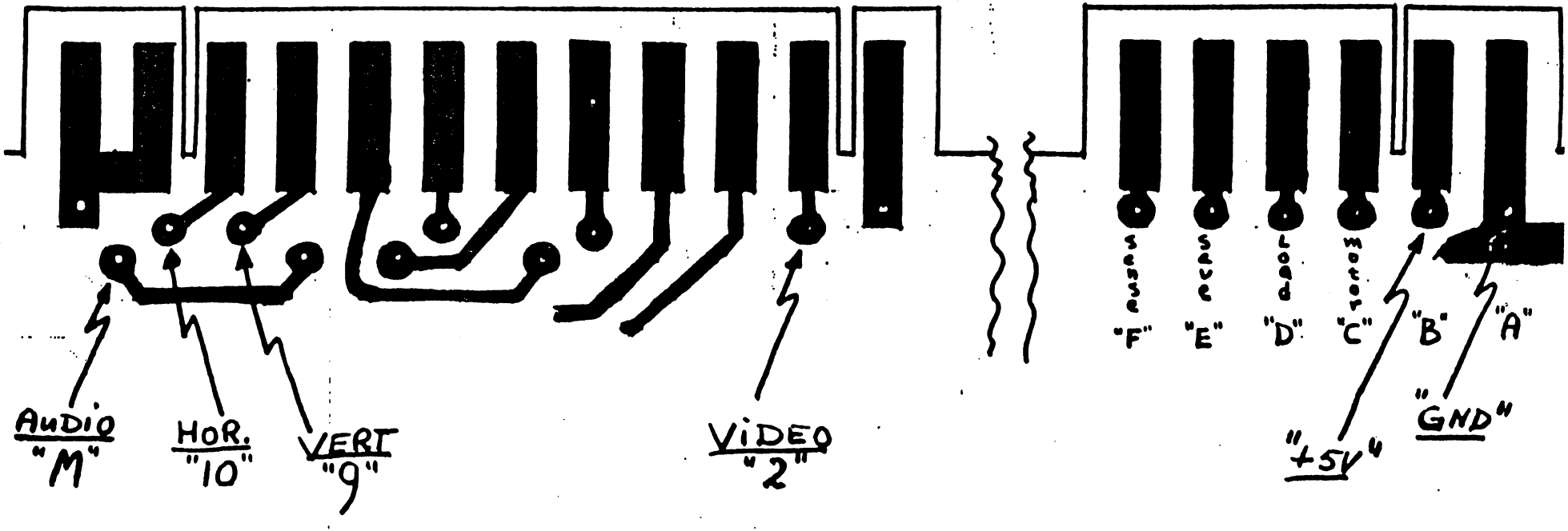
* Note: Only required, when "DUMP" is used.

AV-8101
-2001-

PARALLEL USER PORT

CASSETTE PORT

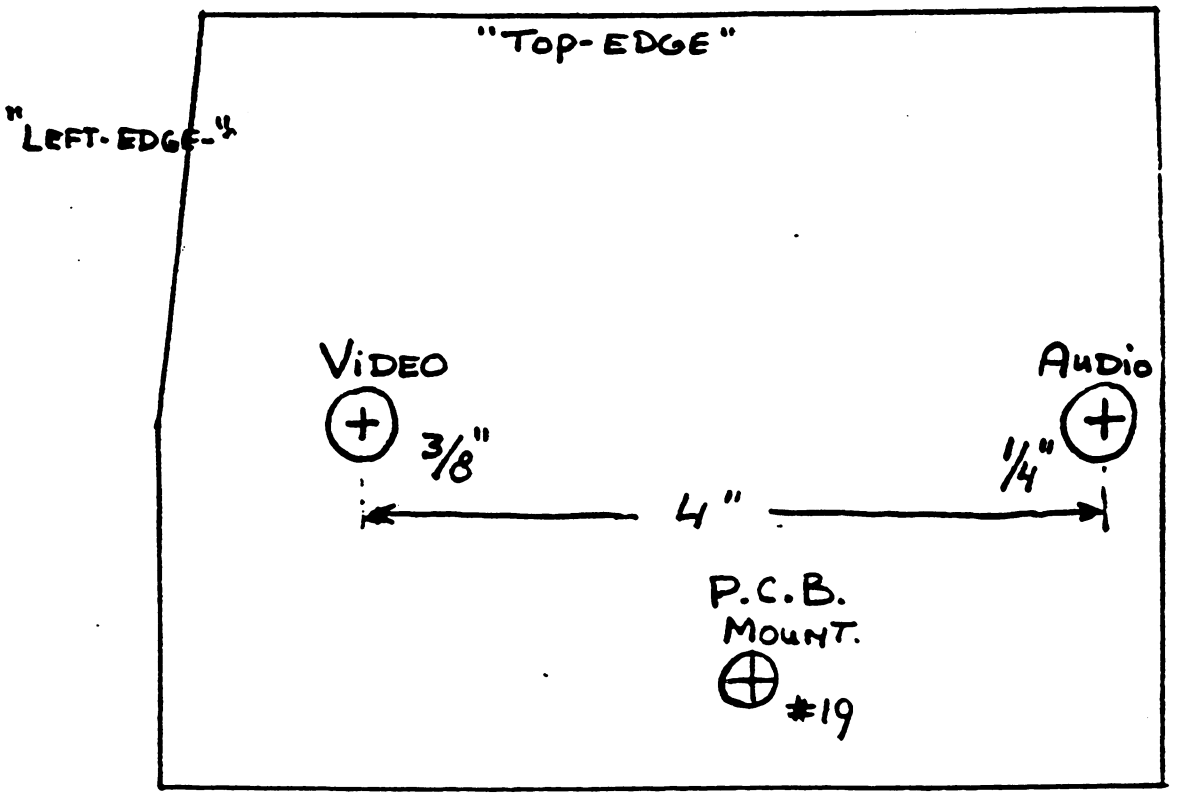
PET2001 - "COMPONENT SIDE" VIEW



1. REMOVE SOLDER FROM THE SIX INDICATED PADS.

2. CONNECT AND SOLDER APPROPRIATE LEADS FROM THE A/V-INTERFACE BOARD TO THESE PADS.

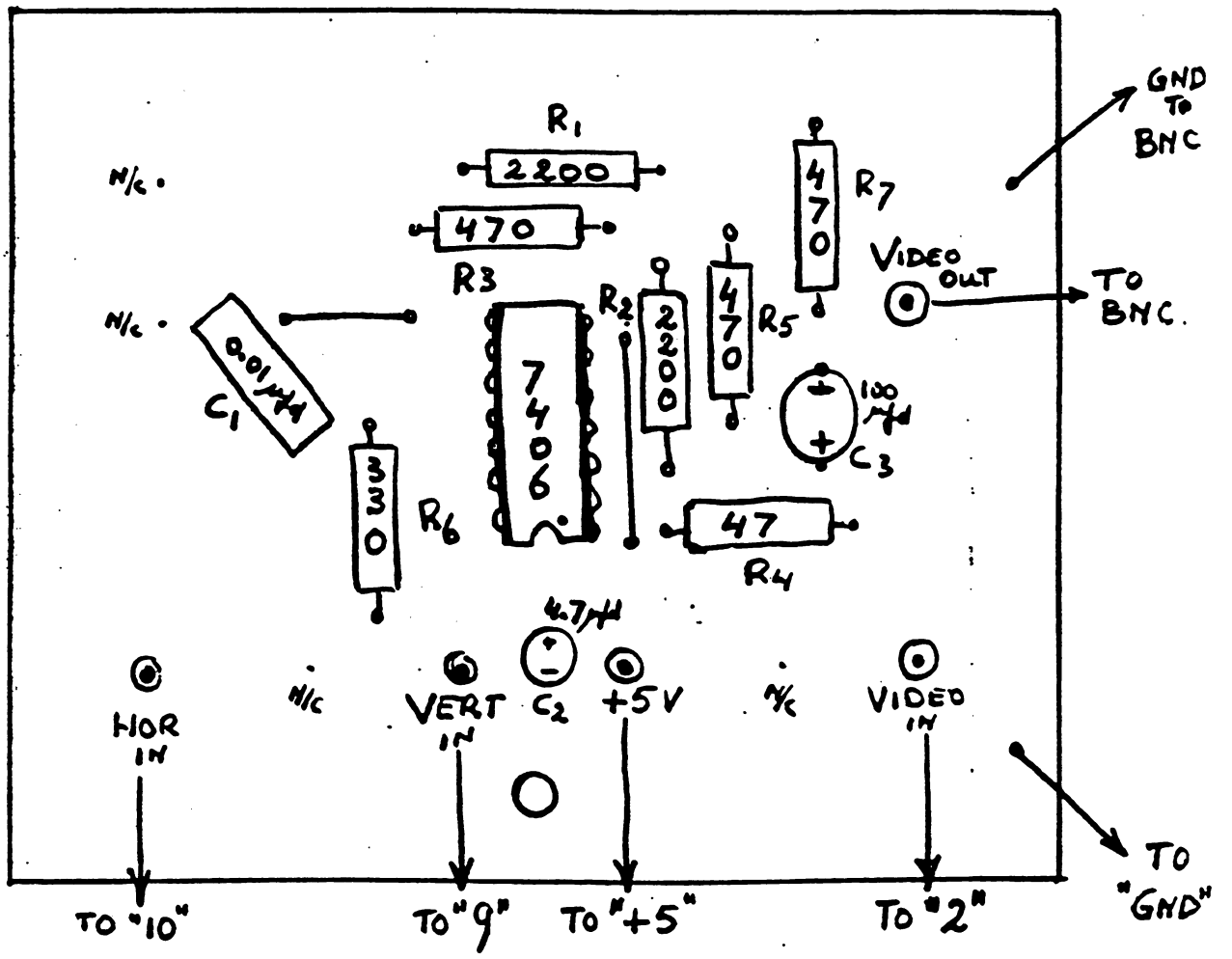
3. TRIM EXCESS LEADS AND CHECK FOR SHORTS, ETC.

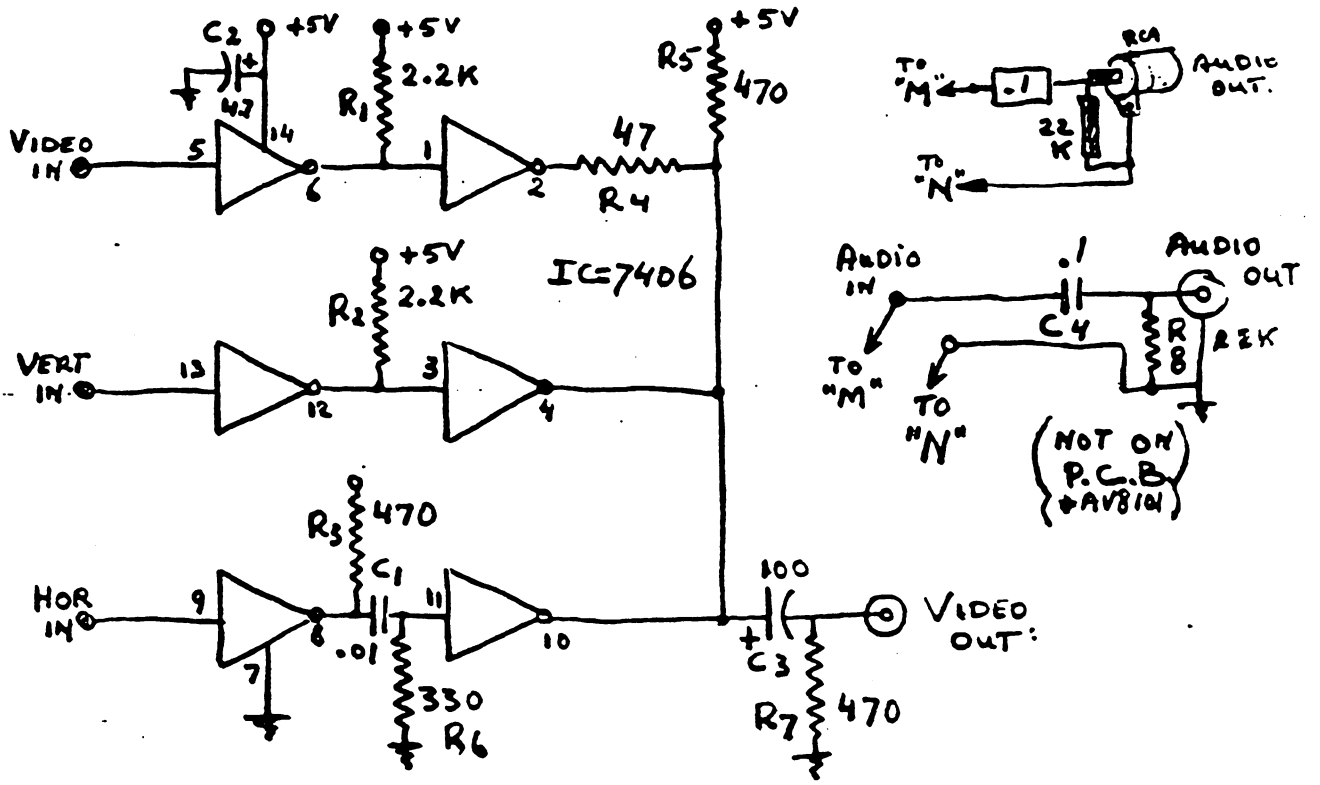


ALTERNATE TEMPLATE FOR "NEW-TYPE"
LOW-BACK - BASE.

AV-8101
-8032-

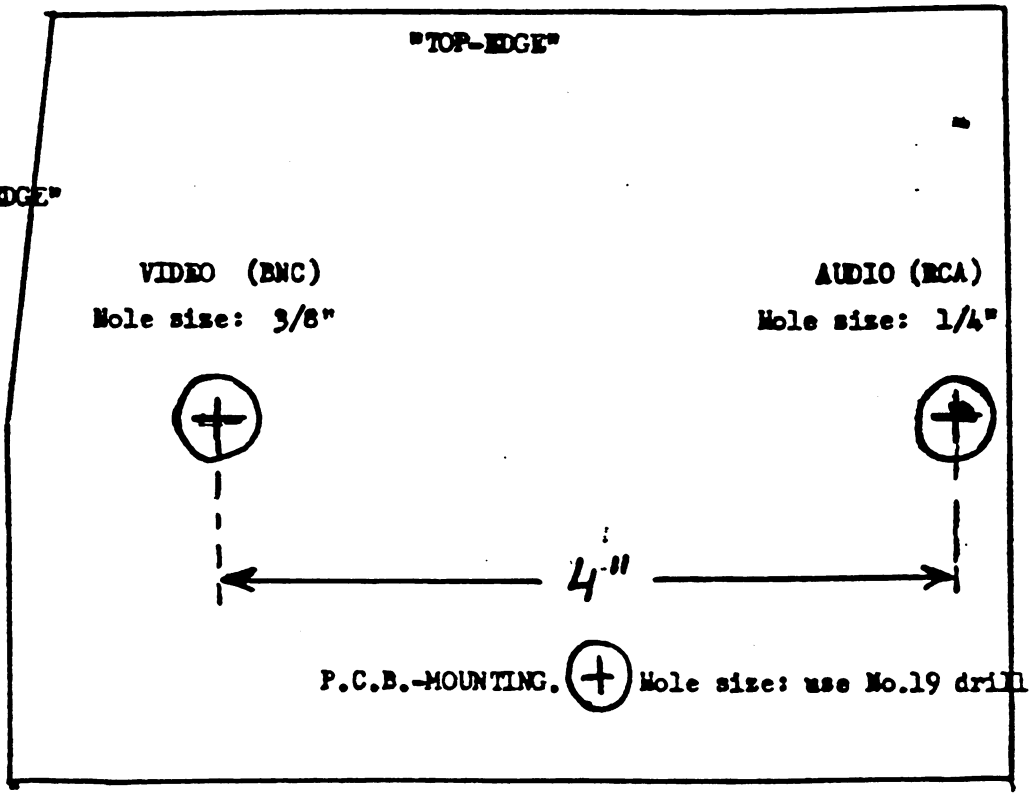
AV-8101-M. INTERFACE BOARD (CBM8032)





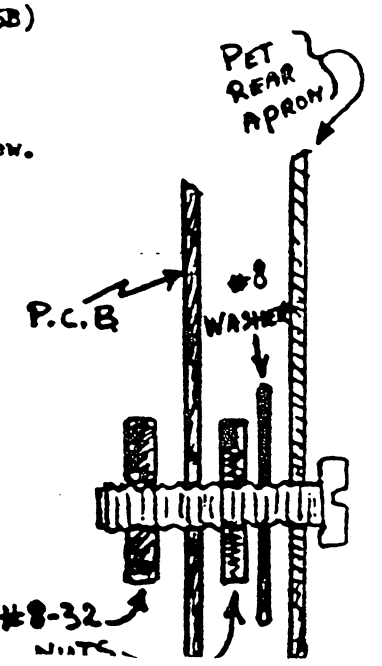
AV-8101-A
-8032-

CBM 8032 REAR APRON DRILLING TEMPLATE



COMPONENT-REQUIREMENTS:

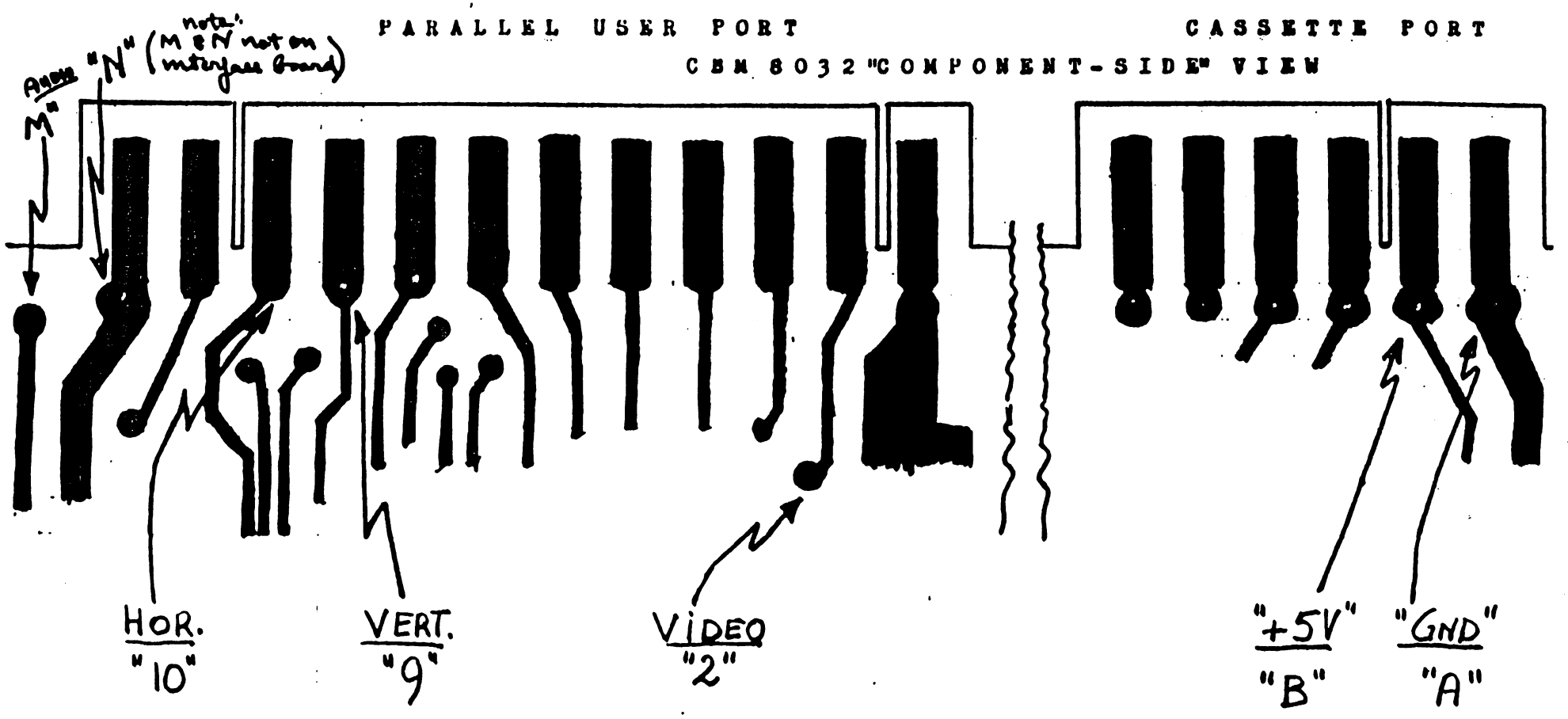
BNC-CONNECTOR	:	Amphenol 31-236 (UG/U-625B)
RCA-CONNECTOR	:	Switchcraft 3501-FP.
GROUND LUG	:	Amphenol 31-759.
HARDWARE	:	1 8-32 X 1/2" machine screw.
		1 No. 8 washer.
		2 8-32 nuts.



PARALLEL USER PORT

CASSETTE PORT

CBM 8032 "COMPONENT-SIDE" VIEW

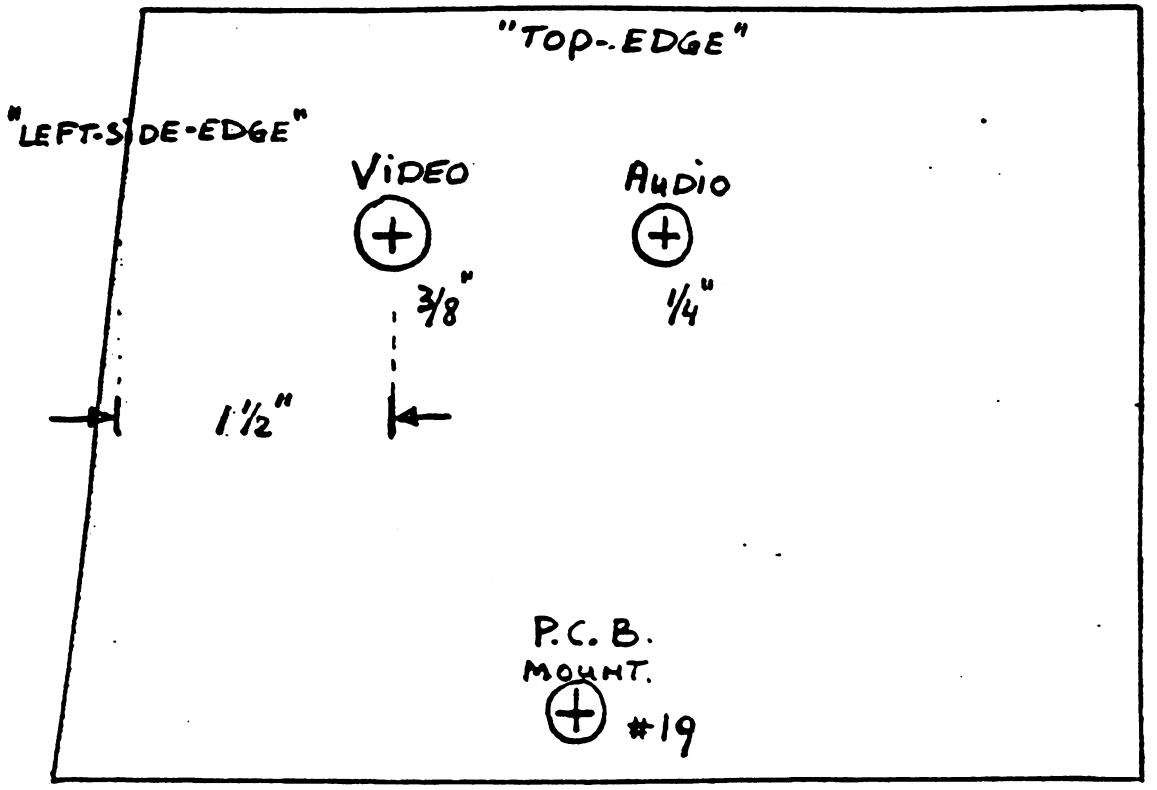


1. REMOVE SOLDER FROM PADS "A" "B" "2" "N" "M"

2. CONNECT AND SOLDER APPROPRIATE LEADS FROM THE INTERFACE BOARD TO THESE PADS, AS THERE ARE NO PADS FOR "9" AND "10", BE VERY CAREFUL WHEN ATTACHING THESE WIRES TO THE ENDPART OF THE CONTACT-FINGERS.

3. TRIM EXCESS LEADS AND CHECK FOR SHORTS, ETC.

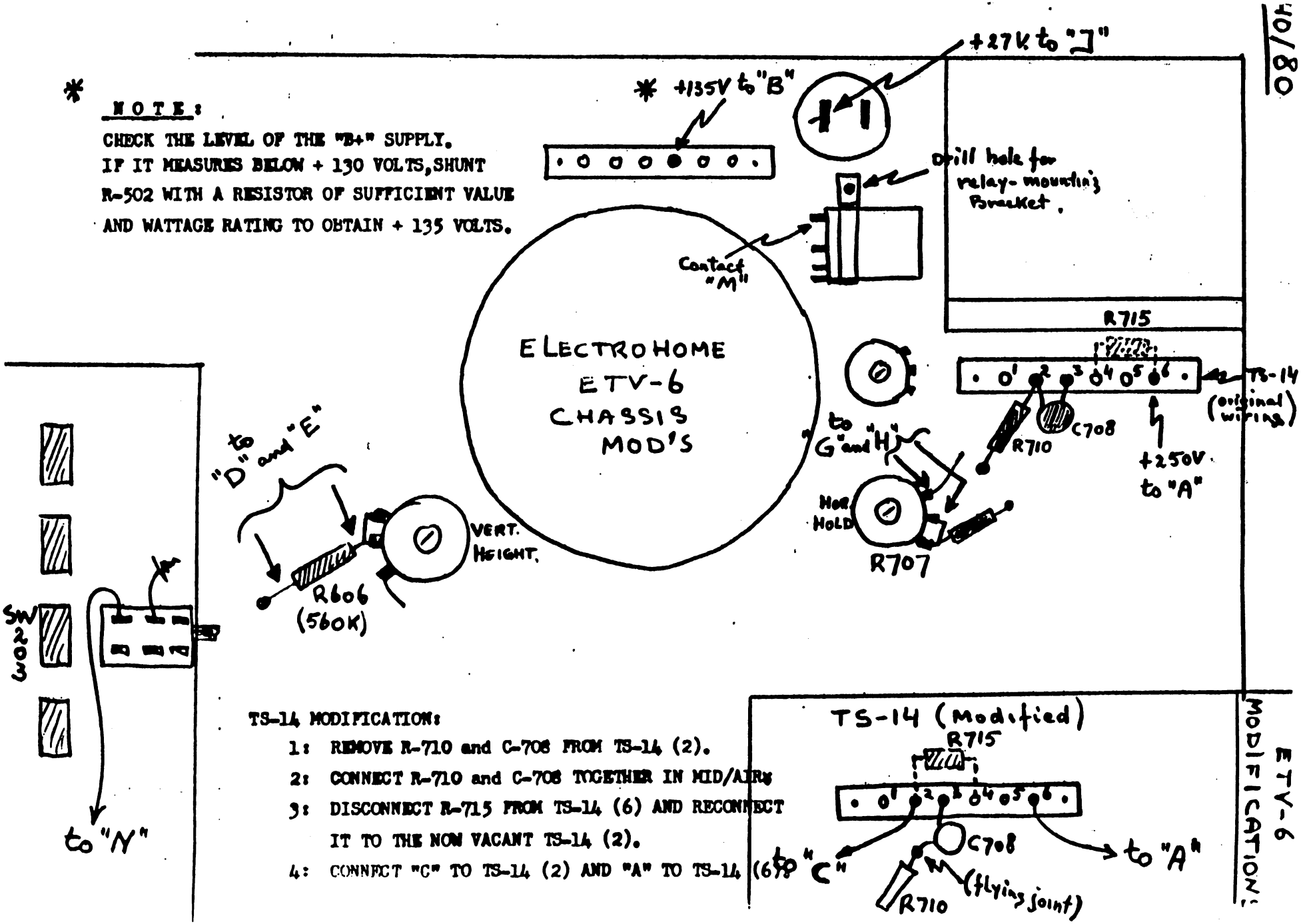
AV-8101-M
-8032-



ALTERNATE TEMPLATE FOR "OLD-TYPE"
HIGH-BACK-BASE

* **NOTE:**

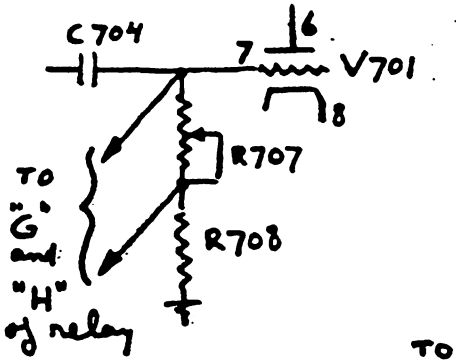
CHECK THE LEVEL OF THE "B+" SUPPLY.
IF IT MEASURES BELOW + 130 VOLTS, SHUNT
R-502 WITH A RESISTOR OF SUFFICIENT VALUE
AND WATTAGE RATING TO OBTAIN + 135 VOLTS.



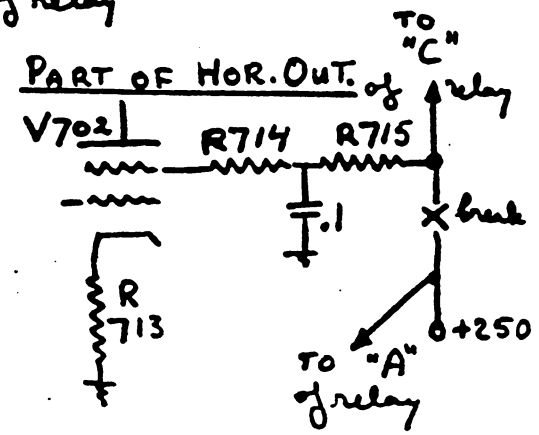
40/80

ELECTROHOME ETV-6 MODIFICATIONS

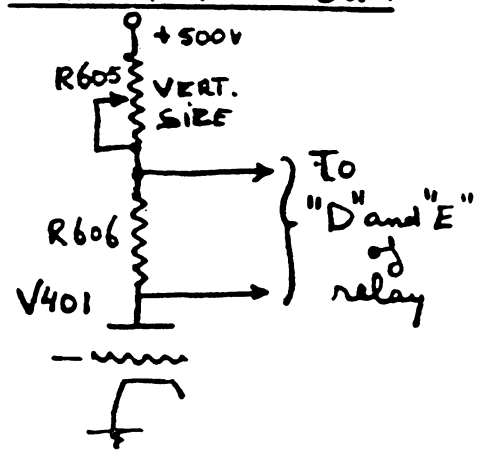
PART OF HOR. OSC.



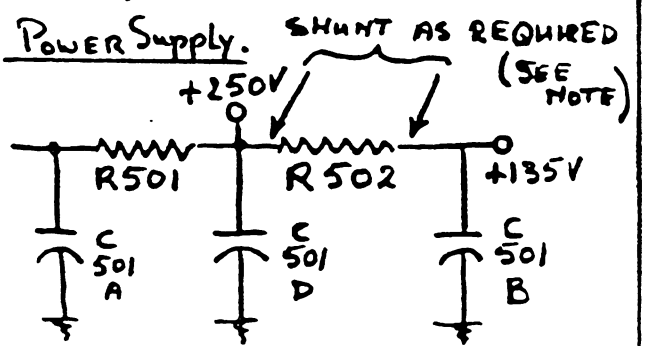
PART OF HOR. OUT.

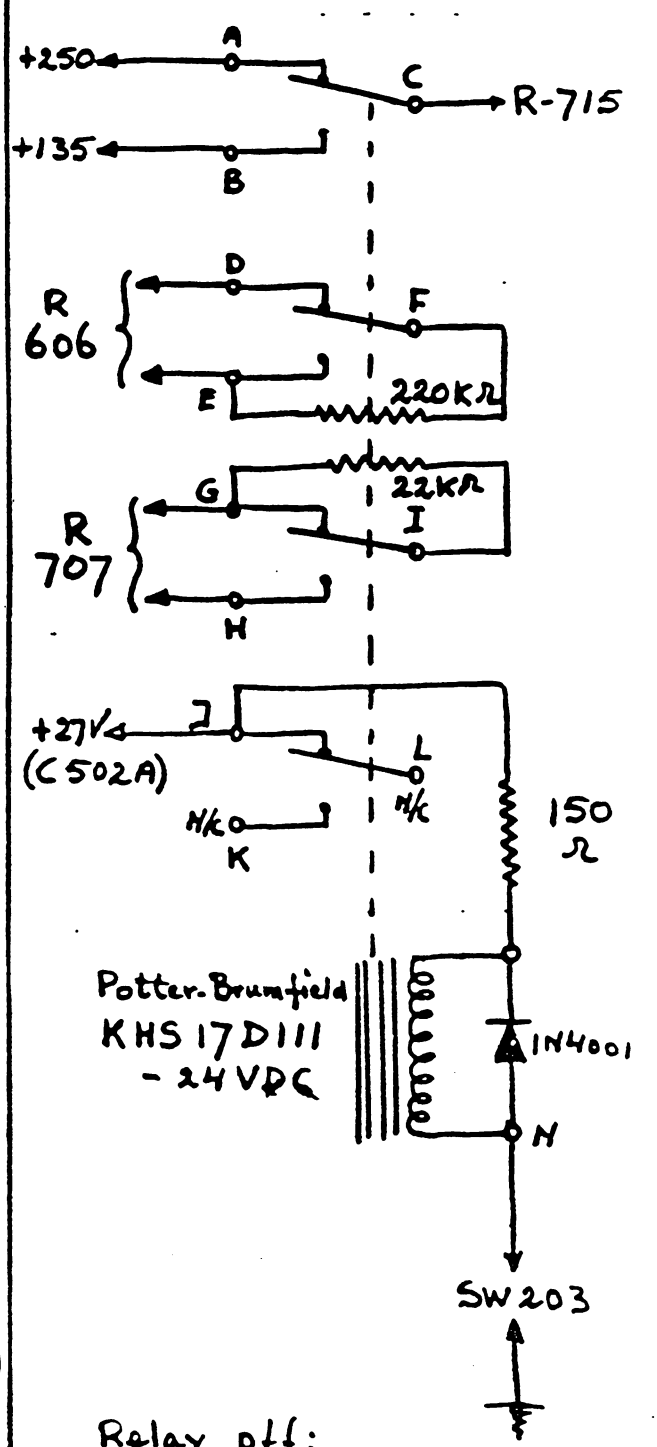


PART OF VERT. OUT.



POWER SUPPLY.





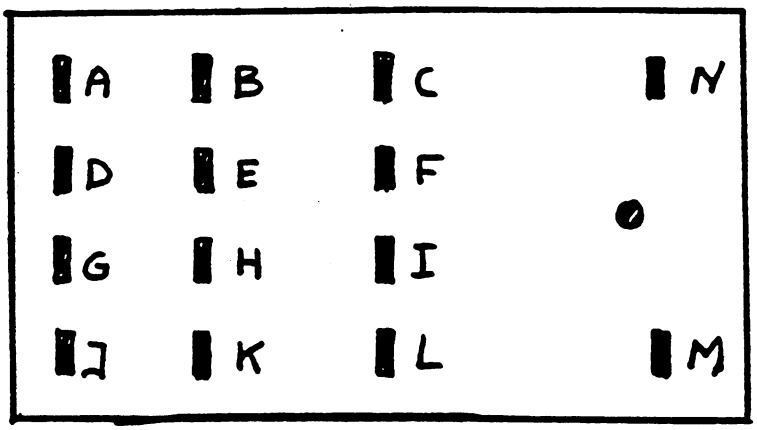
Relay off: (as shown)
NORMAL operation

Relay on: "80-column" operation

40/80

ELECTROHOME ETV-6
MODIFICATIONS

RELAY WIRING.

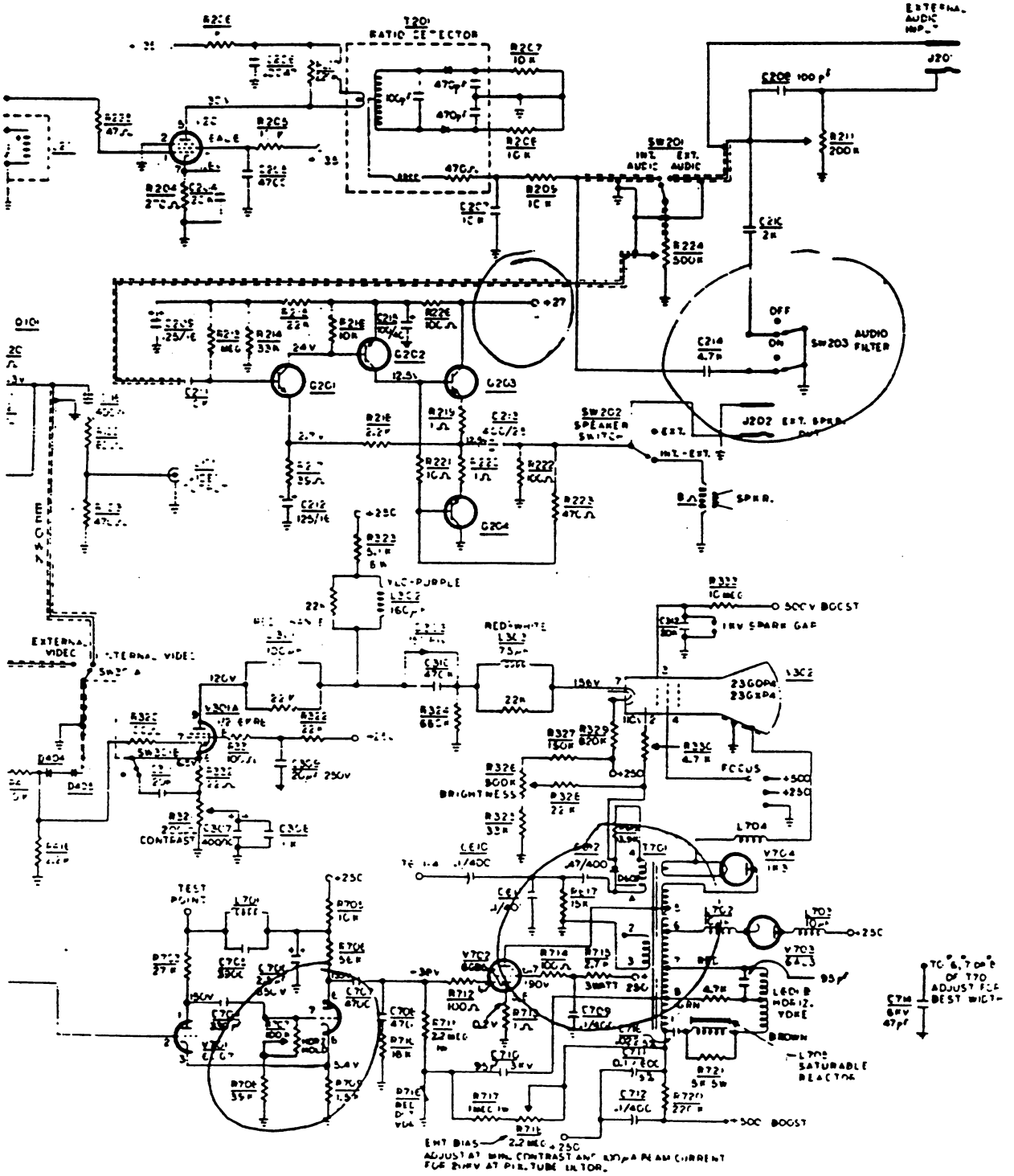


RELAY - SOLDER-VIEW

1: INSTALL 220 K.OHM RESISTOR BETWEEN "E" AND "F".
2: INSTALL 22 K.OHM RESISTOR BETWEEN "G" AND "I".
3: INSTALL DIODE (1N4001) BETWEEN "N" AND "M". (Observe polarity)
4: INSTALL 150 OHM RESISTOR BETWEEN "M" AND "J".
5: CONNECT "A" TO + 250 VOLT SUPPLY.
6: CONNECT "B" TO + 135 VOLT SUPPLY.
7: CONNECT "C" TO TS-14 (2). (Note: TS-14 should have been modified as per) (page 1 instructions.)
8: CONNECT "D" AND "E" ACROSS R-606.
9: CONNECT "G" AND "H" ACROSS R-707.
10: CONNECT "J" TO + 27 VOLT SUPPLY.
11: DISCONNECT ALL WIRING FROM SW-203. (Audio-filter-switch on front panel)
12: CONNECT CENTER CONTACT OF SW-203 TO GROUND.
13: CONNECT "ON" CONTACT (rear connection) OF SW-203 TO "N" OF RELAY.
14: SOLDER ALL CONNECTIONS AND INSPECT WIRING FOR PROPER DRESSING.

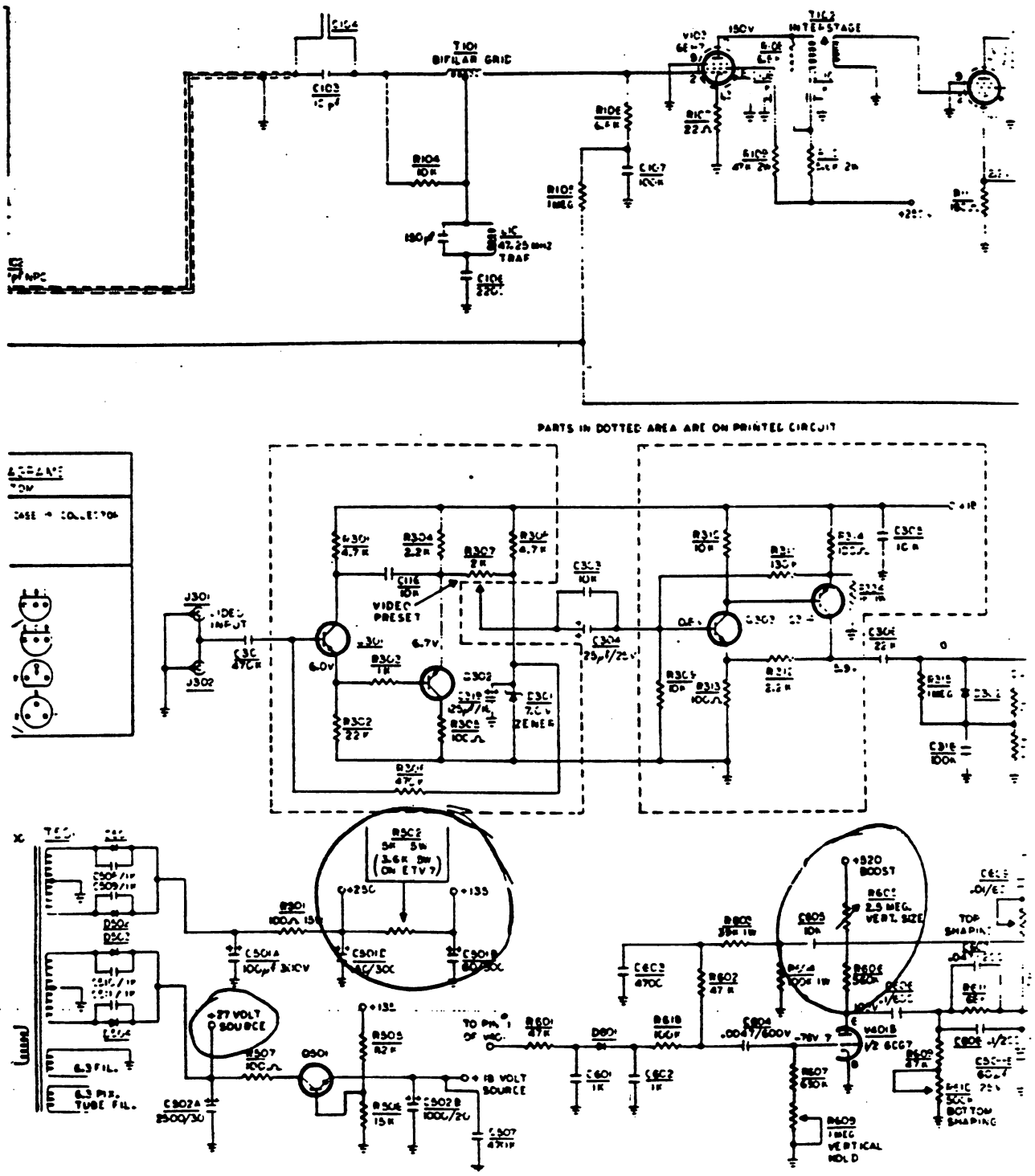
40/80

ELECTROHOME ETV-6 MODIFICATIONS

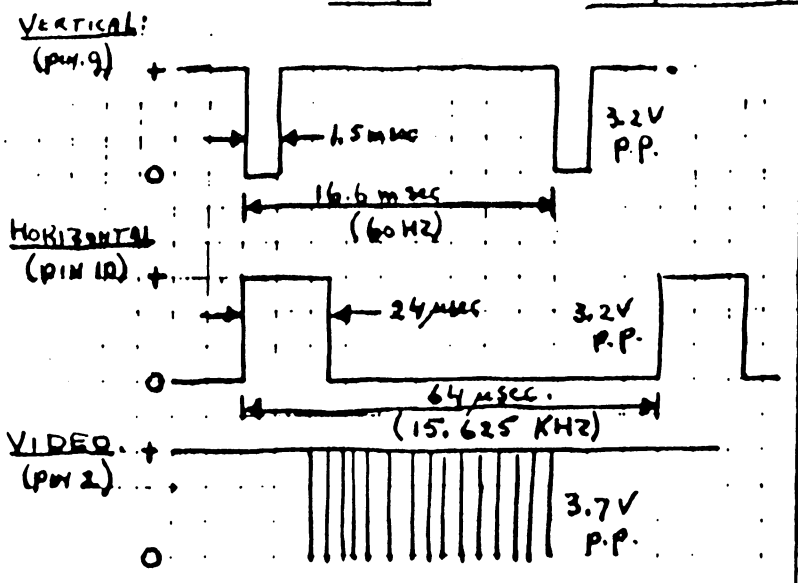


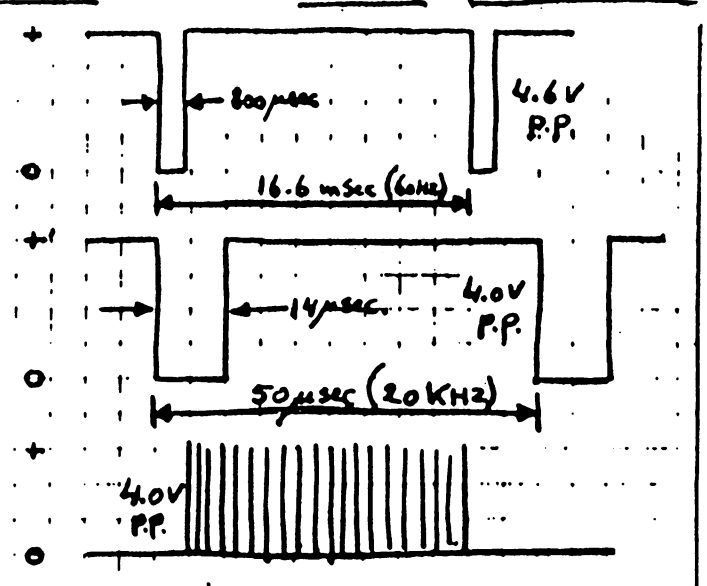
40/80

ELECTROHOME ETV-6 MODIFICATIONS

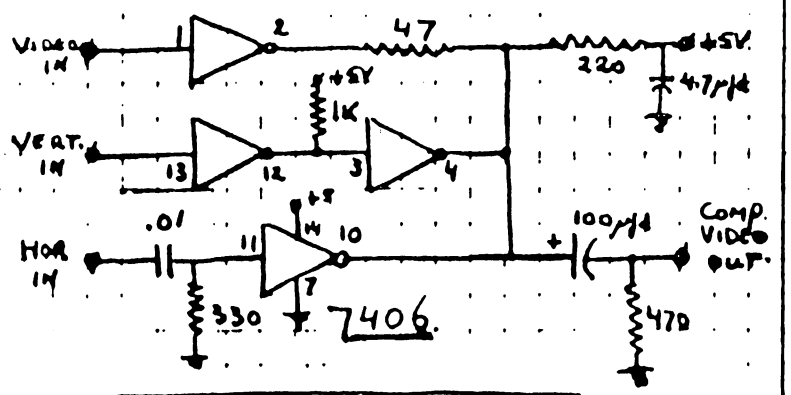


2001 ← OUTPUT-WAVEFORMS → 8032 | FEBR. 16/81. | T.E.



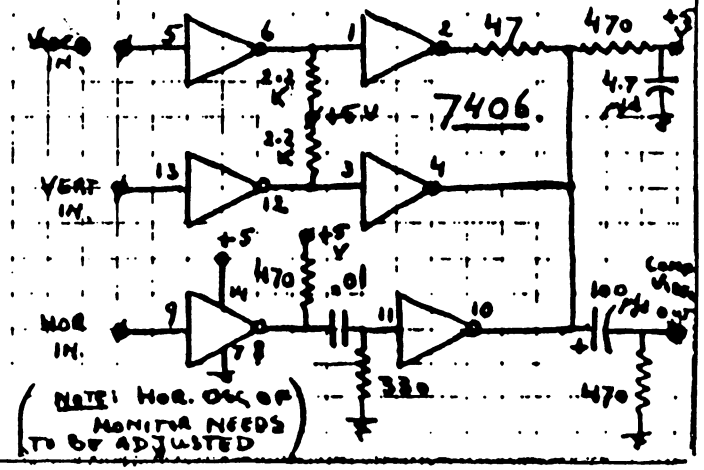


2001 - VIDEO-INTERFACE



M.S.S.B. - A/V - DEPT. - TORONTO

8032 - VIDEO-INTERFACE



NEW DIAL-A-ROM models from KOBETEK SYSTEMS LTD.

It has now been over a year since we first introduced the Dial-a-ROM. At that time, there were not too many chips around (WORPRO and Toolkit). In the past months, however, there has been a steadily increasing number of various utility and protection ROMs. There is also room in the VIC expansion board for two 24-pin ROMs. We suspect therefore that the nunber of (P)ROMs will grow at an increasing pace.

The Model 6H Dial-a-ROM has room for 6 ROMs. It comes equipped with a flat ribbon cable, at the end of which is a 24-pin DIP plug that fits into one of the two empty sockets on the BASIC 4.0 machines (4016, 4032, 8032, VIC). Because of its design (the 6-socket board is housed in a plastic cabinet that sits next to the machine), the flat ribbon cable will not interfere with any expansion boards that you may want to fit on top of the motherboard. The only thing that goes into the computer is the plug. The model 6H has a 6 position rotary switch on the front, which allows you to select one of the 6 ROM positions inside. Note that only one ROM is selected at one time. Also all ROMs inside the unit must be addressed at the socket into which you have put the plug (unless you move the plug to the other socket).

The model 33H is for those people that don't need room for 6 ROMs with the same address. It also has 6 sockets, but they are electronically grouped in two groups of 3 (or in a group of 2, and one of 4 - the user decides by means of a jumper). The 33H also has a flat ribbon cable with a dipplug, and has a second cable with a socket at the end. The dipplug goes into one of the ROM sockets on the computerboard, and the cable socket goes into the other. The 33H has 2 6-position rotary switches, one to control each of the sockets. It permits two ROMs to be selected at the same time, one in each of the ROM sockets of the CBM.

The model 66H is for those who don't need room for 6 ROMs with the same address, and also have no need to have two ROMs active at the same time. This is probably most of us. The 66H has a dipplug cable and a socket cable like the 33H, i.e. one unit will do for both of the CBM sockets. A toggle switch on the front allows you to select one of the two CBM sockets. You then dial to the position of the socket in the 66H which contains a ROM addressed at the selected CBM socket. The 66H offers the user complete freedom in the mix of ROMs in the unit to a maximum of 6.

In the works: The 11H, a daisy-chained model for those who need more than 6 ROMs in the same socket. It is a pair of 6H's tied together, with its own powersupply.

Prices (Canadian funds) - Please add $ 5.00 airmail post

Model 6H	115.00
Model 33H	135.00
Model 66H	135.00

For more information contact Sieg Deleu at Kobetek Systems, R.R.#1, Wolfville, Nova Scotia, 902 542 9100.

JOYSTICK/KEYBOARD ROUTINE _____ David Hook, Barrie Ont.

Purpose:

Many programs, especially games, allow the user to control movement using the numeric keypad. Usually, pressing '8' means 'up', '2' means 'down', '4' means 'left' and '6' denotes 'right'. Diagonal movement is also permitted with the 1,3,7 & 9 keys. The middle of the keypad, 5, signifies no movement or no change in movement. In games, the 'SPACE' sometimes indicates a 'fire' button.

Since I have produced a joystick interface, it would be great to have a single program work for both joystick and keyboard input.

The program will work on all three ROM versions and be easily incorporated into new and older programs.

Description:

Combining a test to screen out all other keys, and decoding the keys takes a fair bit of coding. Fast animation is sometimes restricted by the processing time involved.

Wouldn't it be useful to have a simple 12-line program to handle all of this. The routine should be initiated with a simple command and ignore all but the "desired" keys. Such as...

```
50 N=USR(0):IF N=0 THEN 2000
60 ON N GOTO 100,200,300,400,500,600,700,800,900
```

Thus, the variable N takes on values 0-9 only. Zero corresponds to the 'SPACE' key. All the others can be handled as their proper direction. No other keys will affect 'N'. N=5 when no key has been sensed. This could be programmed to mean no movement, or as no change in direction.

The same input to 'N' is made when the joystick is connected. Now the '0' reflects the 'button' being pressed.

JS/KB LOADER:

Enter the 14-line program as in the listing provided. Save it immediately, as a precaution against an accidental "crash".

RUN the program.

The program loads a short Machine Language routine (93 bytes) into the first cassette buffer. Even if you are loading from Cassette#1, the routine should be safe there until you load another program. The USR vector will be set

according to your answer to the "using joysticks" question.

Any time you enter an expression...

variable=USR(0)

...the routine will be invoked and the variable will assume the value 0-9, as above.

Line 13 in the program is a quick test of the program. Hold down each number key as it's running. Hold down the 'SPACE' key also. Note the numbers appearing on the screen as it runs.

Both Line 0 and Line 13 aren't necessary when you come to use the program. It shouldn't be too difficult to modify existing Basic programs to incorporate the dual functions.

While you are at it, consult Compute #4 or Best of Pet Gazette. I've had only a passing acquaintance with a soldering iron. My joystick interface works great with the above routine. Trying the four Cursor joystick-compatible programs is another challenge.

Modifying five versions of Space Invaders (old, 2.0 normal, 2.0 fast, 4.0 normal & 4.0 fast) is another tale. You can find these at any of three Toronto PET dealers.

Note to hackers: I've selected the 'SPACE' as the 'fire' button. If you want another, change the ninth data item from '32' to the ASCII value of the key of your choice.

Source code for the MAE or ASSM/TED assembler is also included.

```
                                .CE
                    ;*****************************************************
                    ;*                                                   *
                    ;*              JOYSTICK/KEYBOARD ROUTINE            *
                    ;*                                                   *
                    ;*(C) DAVID A. HOOK, 58 STEEL STREET, BARRIE, ONTARIO *
                    ;*                                                   *
                    ;*               L4M 2E9   (705) 726-8126            *
                    ;*                                                   *
                    ;*****************************************************
                    ;
                    ;
                    USRPORT     .DE $E84F INPUT FROM USER PORT FOR JOYSTICK
                    GETBYT      .DE $FFE4 GET A BYTE
                    KEY1        .DE $0203 WHICH KEY--ROM 1.0
                    KEY2/4      .DE $0097 WHICH KEY--ROM 2.0 & 4.0
                    INTFLP1     .DE $D278 INTEGER TO FLOATING--ROM 1.0
                    INTFLP2     .DE $D26D INTEGER TO FLOATING--ROM 2.0
                    INTFLP4     .DE $C4BC INTEGER TO FLOATING--ROM 4.0
                    ;
                    ;
                                .BA $027A STORE IN FIRST CASSETTE BUFFER
                    ;
027A- 20 E4 FF      KEYBD       JSR GETBYT
027D- C9 30                     CMP #'0       ;ZERO KEY?
027F- F0 0F                     BEQ NOCHG
0281- C9 20                     CMP #'        ;SPACE KEY?
0283- D0 02                     BNE NOTSPC
0285- A9 30                     LDA #'0       ;SPACE=BUTTON
0287- 38            NOTSPC      SEC
0288- E9 30                     SBC #'0
028A- 30 04                     BMI NOCHG     ;WRONG KEY
028C- C9 0A                     CMP #10
028E- 30 02                     BMI HAVKEY
0290- A9 05         NOCHG       LDA #5        ;ENTER KEY VALUE OF '5'
0292- A8            HAVKEY      TAY  ;SAVE IT IN R(Y)
0293- A2 FF                     LDX #255      ;= NO KEY PRESSED
0295- AD C6 02                  LDA ROMTYPE   ;1.0=$FF 2.0=$00 4.0=$01
0298- 10 06                     BPL NOT1      ;FOR ROM 2.0 & 4.0
029A- 8E 03 02                  STX KEY1      ;MEANS NO KEY PRESSED
029D- 4C B4 02                  JMP USER
02A0- 86 97         NOT1        STX *KEY2/4   ;MEANS NO KEY PRESSED
02A2- 4C B4 02                  JMP USER
                    ;
                    ;
02A5- AD 4F E8      JOYSTK      LDA USRPORT   ;GET JOYSTICK BIT PATTERN
02A8- 4A                        LSR A         ;DIVIDE BY 16
02A9- 4A                        LSR A
02AA- 4A                        LSR A
02AB- 4A                        LSR A
                    ;
02AC- 2D 4F E8                  AND USRPORT
02AF- A8                        TAY  ;KEEP VALUE
02B0- B9 C7 02                  LDA JS.TAB,Y          ;USE AS INDEX TO KEY VALU
02B3- A8                        TAY  ;LO BYTE FOR INTEGER CONV
                    ;
                    ;
```

```
                0720 ;
2B4- A9 00      0730 USER       LDA #0          ;HI BYTE FOR INTEGER CONV
2B6- AE C6 02   0740            LDX ROMTYPE     ;1.0=$FF 2.0=$00 4.0=$01
2B9- 10 03      0750            BPL TWO/FOUR
2BB- 4C 78 D2   0760            JMP INTFLP1     ;TO FLOATING FOR 1.0
2BE- D0 08      0770 TWO/FOUR   BNE FOUR
2C0- 4C 6D D2   0780            JMP INTFLP2     ;TO FLOATING FOR 2.0
2C3- 4C BC C4   0790 FOUR       JMP INTFLP4     ;TO FLOATING FOR 4.0
                0800 ;
                0810 ;
                0820 ;
2C6-            0900 ROMTYPE    .DS 1           ;1.0=$FF 2.0=$00 4.0=$01
                0910 ;
2C7- 05 05 05   1000 JS.TAB     .BY 5 5 5 0 5 7 9 8 5 1 3 2 5 4 6 5
2CA- 00 05 07
2CD- 09 08 05
2D0- 01 03 02
2D3- 05 04 06
2D6- 05
                1010 ;
                9995            .EN
```

ABEL FILE: [/ = EXTERNAL]

```
'USRPORT=E84F            /GETBYT=FFE4            /KEY1=0203
'KEY2/4=0097             /INTFLP1=D278           /INTFLP2=D26D
'INTFLP4=C4BC            KEYBD=027A              NOTSPC=0287
IOCHG=0290               HAVKEY=0292             NOT1=02A0
IOYSTK=02A5              USER=02B4               TWO/FOUR=02BE
:OUR=02C3                ROMTYPE=02C6            JS.TAB=02C7

'/0000,02D7,02D7
```

What Hardware do I Have?

Jim Butterfield, Toronto

Users used to tell their PET/CBM machines apart by the size and shape of the keyboard, or by the message that was displayed when power was turned on. That doesn't work too well any more. New keyboards can be fitted to old machines; new ROM sets can be plugged in; and even the green screen/white screen isn't much of a hint any more.

Although I'm not a hardware man, I often get calls from users who want to know about some aspect of their machine. I try to establish which machine they are talking about. Commodore may have a much more official checklist for their hardware configurations: but here's the set I use. Perhaps readers can suggest other differences that are worth while knowing.

Items 1 and 2 are pretty obvious: how many columns on the screen - 80 or 40 - and what type of keyboard. I'm not concerned with tiny versus full-sized keyboards; rather, is it a simple graphics keyboard or a full-scale business ASCII layout? The easy way is to look at the top row, above the alphabetic letters: are there numbers along the top or just symbols?

The next question: does your PET have original architecture or the more recent layout? The tipoff here is the connector on the right-hand side of your machine. If you see an edge-connector - a connector with copper "fingers" extending from the board inside - you have the original machine board. On the other hand, if you can see a series of upright pins when you peer through the right hand slot, you have a more recent board. Another way of telling the same thing is to type POKE 59409,52. If the screen goes blank when you press RETURN, you have a unit with original architecture. Here's another characteristic of the early machines: if you type FOR J=1 TO 1000:POKE 32768,0:NEXT J there will be a lot of "snow" on the screen for a few moments; newer machines don't have this problem.

The next thing to test is the screen writing speed. Clear the screen, and type on the top line or two: TI$="000000":FOR J=1 TO 600:PRINT "A"; : NEXT J: PRINT TI$ and press RETURN. You'll get a lot of letter A's across the screen, followed by a number. If the number is 000002 or less, you have a fast screen machine. If it's bigger, you have a slower screen unit. Don't worry - the speed difference is only seen when writing characters to the screen, and you can't read that fast, anyway.

There are a couple of minor cosmetics that are sometimes worth noting. Do you know which cassette unit plugs into the back connector? Is it cassette number 1 or 2? There's a difference in board wiring either way. And finally, if you like rummaging around the inside of your PET, how many pins do you have on your ROM chips? If you don't know how to spot

a ROM chip, you don't need to know ...
While you have the lid open, count the number of ROM sockets.

There are other hardware differences that you'll find in various PET/CBM machines, but the above are the ones I ask for most often. Technical tyros will be glad to add to the list: what kind of RAM is fitted? How are the ROM sockets decoded? .. and so on.

Many other differences that we notice between machines seem to be hardware, but they're really software. It's quite surprising how different logic can make the machine appear physically different. We'll talk about software, or ROM sets, a little later.

Some people claim that cosmetics make a great deal of difference to a computer's usefulness. I suppose it's part of your state of mind. I've seen PETs with racing stripes and others with pink polka-dots. If the owners feel that they can write better programs that way, good. I'm thinking of painting one of mine puce-coloured in the hopes that it will keep the cat away...

What Software System?

There are (at time of writing) three general styles of PET/CBM software ROMs. ROM stands for Read Only Memory - their programs are set at the factory and cannot be changed. That's OK - few of us have the talent and/or ambition to change the internal workings of our PET/CBM system, and even if we did we'd lose one of the great advantages of the home computer: the ability to exchange programs with others. The ROM programs, being pre-written and burned in, are there the moment we turn the power on.

I call the three generations of Basic: Original, Upgrade, and 4.0. The first two generations had confusing number systems: some Commodore divisions called Original ROM sets Basic 1.0; others called the same thing Level 2 Basic. When the Upgrade system arrived, the numbers changed to Basic 2.0 and Level 3 so as to make the confusion one hundred per cent. By the time 4.0 Basic came along everybody synchronized, and the machine prints BASIC 4.0 to end the problem once and for all.

Within each Basic version, there are small differences to accommodate variations in the hardware. The Business or ASCII keyboard - the one with numbers across the top row - needs to be scanned in a different way than the graphics keyboard; and 80 column screens must be worked in a style that differs from the 40 column display. These differences are reflected by changing one ROM out of the set to allow for the configuration desired. The other ROMs in the set are the same regardless of hardware.

The first Basic came as part of the Original ROM set. It had a lot of limitations. You couldn't put more than 256 items into an array; you couldn't do a successful IEEE-488 input; you had no machine language monitor; tape data files had potential problems. Most users breathed a sigh of relief when the newer Upgrade system became available.

If you have Original ROM Basic, your PET will power up with the message: *** COMMODORE BASIC ***. Note the asterisks: they are the signal that you have the Original system.

If you have this early Basic, it's worth while thinking about moving up by obtaining a replacement ROM set. You'll get technical benefits. More important, you'll be joining the mainstream of PET/CBM users and be better equipped to exchange ideas and programs.

Upgrade ROM solves the above limitations. Users with Upgrade ROM will see ### COMMODORE BASIC ### when they turn the power on. Whether you call those symbols pound signs, number symbols or hash marks they clearly flag Upgrade ROM.

Users with original hardware can refit their machines to Upgrade ROM, but they will have trouble in taking the next step to 4.0. It's not just that they are missing the ROM socketing to plug everything in: they would also find that screen "noise" would start to give trouble. Basic 4.0 doesn't politely wait for the screen to be ready before delivering new information ... characters are slapped in at full speed. Newer machines won't see any problem, but the original boards may end up with a screen that looks like a snowstorm.

The newest Basic so far is 4.0 and it's easy to spot: the screen announces the number. The changes here are useful, but not essential. You get new commands for disk: things like DLOAD, CATALOG and SCRATCH. You get somewhat better file handling; and that great time-waster, garbage collection, has been speeded up so that it is no longer annoying. In many respects, the 4.0 improvements are largely cosmetic; they support ease of use rather than eliminating fundamental road blocks.

There's great compatibility between the various versions of Basic, especially between Upgrade and 4.0 ROMs. Each user tends to exploit the features he is given, however, so that programs on a more recent model may not be able to time-travel back to earlier versions.

Still, they are all PETs. They all have that style ...

The Fat 40
A New Generation 40 Column PET

Jim Butterfield
Toronto

There's a new kid on the block. It arrived without any fanfare or advanced publicity: the Fat 40 - a large-screen 40 column PET.

The most noticeable feature of the fat 40 is its big 12-inch screen. You'd almost think you had an 8032 80-column computer until you look at the keyboard and notice that the keyboard layout is graphics rather than ASCII business. When you turn it on, the screen characters come up fat and crisp.

I'd heard that Commodore would be fitting 12-inch screens to their 40-column computers, and had thought that this would be only a screen change. Most of us, I think, were unprepared for a noticeably new machine with new logic. The logic board has changed so that it is similar in layout to that of the 8032. More importantly, the program logic has been changed so that we have more than a new size of machine: we have a new style.

The changes are improvements, but we'll have to recover from the shock of having a new system introduced with no advanced fanfare. Most programs which work on the skinny 40's will move to the fat 40 without trouble; I'll try to itemize possible conflict areas later in this article.

A Junior 80?

The fat 40 picks up much of the style of the 80 column machine. In fact, many of the features of the 8032 are now available in 40 columns, including ringing the bell, erasing part or all of a line, and screen tabulation. Windows - the ability to pick out a piece of the screen - and up/down scrolling, are not available.

The user will notice one of the features right away - as he turns the machine on, a bell chimes. Yes, there's an electronic bell in there, and it chimes any time you get near the end of a line. You can make it ring from your program with PRINT CHR$(7). The bell may be turned off or changed by poking the contents of location 1004 decimal - note that this is a different location from that used in the 80-column machine (231 decimal).

Another feature that's hard to miss is the repeating keys. The cursor movement keys, the Space key, and the Insert/Delete key repeats automatically if the key is held down for a moment. Very handy.

The user may switch to Text mode (upper/lower case) with PRINT CHR$(14). This is easier to remember than the POKE address, and gives you another bonus: the screen lines are readjusted to give you somewhat more pleasant text appearace. You may return to graphics with PRINT CHR$(142).

There are commands for clearing all or part of a line; it takes a little dexterity to get them to work right since you must carefully place the cursor before using them. To clear a line up to but not including the cursor position, PRINT CHR$(150). To clear from the cursor to the end of line, type PRINT CHR$(16). This appears to be a error on Commodore's part, since PRINT CHR$(22) does the job on the 8032 and is more consistent. It seems as if someone has confused decimal 16 with hexadecimal 16 (22) in the coding. I would anticipate this being corrected in future ROMs; in the meantime, you can print both characters and create programs which will move gracefully through any such future correction: PRINT CHR$(16);CHR$(22). This Erase-to-End sequence is often very handy: it allows you to clear a line before printing on it.

Screen tabulation is included. It will probably be less useful on a 40-column machine as compared to its 80-column big brother, but it's not hard to use. Set or reset tab positions by positioning the cursor and then printing CHR$(137). You can move to the next tab location when desired by printing CHR$(9).

System organization changes.

Users who fiddle with the innards of the machine - or who have programs that do so - should be aware of certain internal changes in the fat 40.

The keyboard is decoded in a different manner from previous 40-column machines; you should not depend that PEEK(151) will give you the same values for specific keys being pressed. You may still expect that this value will be 255 if no key is pressed - but the other values will have moved around.

The Screen Wrap table, which tells us which rows of the screen are joined together into double lines, is still in the same place. But this means that some of the 80-column variables needed to be moved clear of this area; values previously stored in locations E3 to E8 have been relocated to the area 03E9 to 03EF. You don't need to know all of their functions, but you must leave these new address locations alone, or you'll interfere with screen/keyboard operation.

80-column tyros might wish that all the wide screen features had been implemented on the fat 40. It probably wasn't possible: the two machines need to be organized differently. Users who go looking for the 80-column screen windows or for the input and output vectors will be disappointed: they aren't there.

Summary.

It's an exciting new machine. It has good new features, and should be very convenient. I wish that Commodore had told us it was coming ... but in any case, I'm glad it's here.

The SUPERPET: A First Look

Joseph P. Ferrari
CBM Software Dept.

Introduction

The dawning of a new age in the micromputer industry is fast approaching. The announcement of the SUPERPET Computer by Commodore has demonstrated that a new breed of micros is on it's way. With 5 languages currently available and a whole lot of RAM space, it makes the SUPERPET one of the most versatile computers on the market today.

Although the SUPERPET has not yet been released, I was able to get my hands on one of these beasts for testing and evaluation. Since it is a pre-release version, all information contained in this article is subject to change. There is a lot to be said about the SUPERPET, but I will discuss mostly Waterloo microBASIC and its features.

Powering up the SUPERPET comes up the usual Commodore Basic, but by flicking a switch located on the bottom right side of the computer will bring up Waterloo microSystems menu. All the languages available are boot-loaded from disk, with the exception of the monitor which is resident in rom. Selecting b <return> will load Waterloo microBASIC with 30150 bytes free.

Editor Features

For those of you who frequently use a developer's toolkit, you will be pleased to know that Waterloo microBASIC includes the following features in their editor for the development of programs.

A) DELETE
B) RENUMBER
C) AUTOLINE
D) MERGE
E) STEP

Debugging Programs

Here is one feature that I really like. When a program has been interrupted either by the opperator or due to an error in the program, changes or modifications can be made and execution of the program can continue at the point of interruption or at any loaction desired without the loss of the current variables.

Saving and Retrieving Program Files

Unlike Commodore basic, Waterloo microBASIC gives two options on how a program can be saved and retrieved.

1) STORE "FILENAME"

Will output the program to the disk in a compressed form. Two advantages of saving programs in this manner are:

a) occupies less space on disk
b) faster in storage and retrieval

2) SAVE "FILENAME"

Will save the program as it is typed on the keyboard and will appear on the directory as a sequential file. This method should be used only while the program is under development. The advantage of this mode is that portions of the program can be saved and eventually merged with other programs by specifying a line range.

3) LOAD "FILENAME"

Retrieves program files that were saved with the 'STORE' command

4) OLD "FILENAME"

Will load a program file that was created with the 'SAVE' command. Any existing program in memory is cleared and the new program is then loaded.

5) MERGE "FILENAME"

Adds the program specified to the existing program in memory. Be very carefull when using this command, for any new line number comming in that exists in memory will be deleted and the new line inserted in its place.

String and Variable Representation

If you have ever encountered a point in program development where you can't think of an appropriate name for a variable, in mircoBASIC you won't have such a problem because all string and variable names can have up to 31 characters with upper and lower case representation.

Repetitive Structures

Waterloo microBASIC supports a variety of statements that make coding easier to implement and also much easier to understand. Since we are all familiure with the FOR-NEXT LOOP structure, I will skip it entirely and go on to some you may or may not know.

WHILE-ENDLOOPS

Delimits a group of statements to be executed repetitively if the value of the expression is true (non-zero)

Example-

```
X=15
WHILE X
   PRINT X
   X=X-1
ENDLOOP
```

The example above will execute statements starting with WHILE and ending with ENDLOOP until the value of variable X is equal to zero. As you can see ENDLOOP is similar to saying GOTO only without specifing a line number. As a point of interest, Waterloo microBasic executes much more efficiently with structured loops in comparison with GOTO statements.

LOOP-UNTIL

Delimits a group of statements to be executed repetitively if the value of the expression in the UNTIL statement is true (non-zero), control is then passed to the statement immediately following the UNTIL statement.

Example-

```
X=15
LOOP
   PRINT X
   X=X-1
UNTIL X=4
```

LOOP-ENDLOOP

Defines a group of statements to be repeated infinitly. To exit loop, the STOP key is pressed or by introducing an IF-THEN statement.

Example-

```
X=0
LOOP
   PRINT X
   X=X+1
   IF X=10 THEN QUIT
ENDLOOP
```

Execution of the statements falling within LOOP and ENDLOOP will be repeated until the value of X is equal to 10. Control is then passed to the statement following ENDLOOP.

IF-THEN-ELSE

Most of us accustomed to Commodore Basic are familiar with the IF-THEN statement. MicroBASIC offers an extension to this statement with the ELSE option. The set of statements following ELSE are executed if the condition is false. ENDIF delimits the ELSE statement group.

Example-

```
COUNTER%=1
LOOP
   TESTER% = INT (COUNTER% / 2)* 2
   IF TESTER% = COUNTER%
       PRINT COUNTER%, "NUMBER IS EVEN"
   ELSE
       PRINT COUNTER%, "NUMBER IS ODD"
   ENDIF
COUNTER% = COUNTER%+1
IF COUNTER%= 10 THEN QUIT
ENDLOOP
```

The above example demonstrates the power of the ELSE option with the IF-THEN statement, especially if more than one statement is to be executed in each condition. I must say that utilization of structured coding makes for far more readable code.

Although there are many more features that I have not covered in this article, I hope that it has given you a small taste of what to expect with Waterloo microBasic on the SUPERPET. In the next issue of The Transactor, I will cover the 6809 ASSEMBLER/DEVELOPMENT package plus some technical aspect of the SUPERPET. The SUPERPET has brought an exciting concept to micro computers and to this end I will try and share with you as much as I can prior to its release.